

IN-82-7M
22452/
P-17

Executive Overview and Introduction to the
SMAP Information System Life-Cycle and
Documentation Standards

INTRODUCTION

The purpose of this overview is to provide information on how to use the Information System Life-Cycle and Documentation Standards, (commonly referred to as the Standards). This information will cover description, objectives, key definitions, structure and application of the Standards, and document structure decisions.

DESCRIPTION

The Standards have been created by the Software Management and Assurance Program (SMAP) to provide consistent NASA-wide structures for documenting information systems and their components. It is the responsibility of the appropriate program/project manager to enforce the use of the SMAP Standards and to ensure their use is integrated with other standards available at the NASA centers.

The Standards document consists of five SMAP Standards:

- The Information System Life-Cycle Standard
- The Management Plan Documentation Standard
- The Product Specification Documentation Standard
- The Assurance Specification Documentation Standard
- The Management Control and Status Reports Documentation Standard.

The last four items listed above are collectively referred to as the "Documentation Standards." Therefore, the entire book has been titled:

"Information System Life-Cycle and Documentation Standards."

OBJECTIVES

The purpose of the SMAP Standards is to provide a standard, tailorable life-cycle process, and the structure and formats for documenting products of that process. The Standards are applicable to information systems and their software, hardware, and operational procedures components. The Standards address only life-cycle and documentation structures; not specific management, engineering, technical, or assurance standards and practices. As such, they complement and augment other existing standards, such as detailed hardware engineering and assurance standards used at various NASA centers. The Standards have been written to be methodology independent for general applicability. For example, they are flexible enough to be applied to either object-oriented or structured software design methodologies.

DEFINITIONS

A first step in ensuring consistency is to make sure all readers are speaking the same language. Therefore, one must first become familiar with the terminology of the Standards.

An INFORMATION SYSTEM is any system composed of hardware, software, and operational procedures components required to process, store, and transmit data. Information Systems, within the context of these Standards are software-intensive systems. Conversely, the act of separating an information system into its hardware, software, and operational procedures components (or into subsystems) is called SYSTEM DECOMPOSITION.

The terms hardware and software should be familiar to users of the Standards and will not be defined here. However, the term OPERATIONAL PROCEDURES may be open to interpretation and, therefore, is defined in the Standards as the manual procedures or processes (conducted by the operator of the system) that are an integral part of an information system's operational capability provided to end users. Within the above definition, then, an OPERATOR (one who executes the operational procedures), is internal to the information system, while a USER is external to the system. Operators executing the operational procedures work in conjunction with the hardware and software to provide the operational capabilities of the information system to the user.

An example of this distinction can be seen in a word processing system that employs a spell checker. A system design decision would be whether to have the spell checker available to the user immediately upon request by having it a part of the main program, or whether to have it on a floppy disk, which must be loaded manually into the disk drive each time it is needed. Assuming the later choice was made, the act of loading the spell checker floppy would be an operational procedure performed by an operator. The act of spell checking the document would be performed by the user. It can be seen that the distinction is sometimes a fine line; and, in this example, the operator and user functions probably would be performed by the same person.

The organization obtaining a capability (i.e., a product such as an information system or a service such as quality assurance) is called the ACQUIRER. The organization supplying the capability to an acquirer is called the PROVIDER.

Any information system (or its hardware, software, or operational procedure component) will require four distinct types of documentation. These are:

- A Management Plan document
- A Product Specification document
- An Assurance Specification document
- A Management Control and Status Reports document.

These four documents are referred to in the Standards as a DOCUMENTATION SET. For relatively small projects, there will be literally only four documents required for the management, development, assurance, and feedback reporting for the product and associated processes.

For large projects, some sections of the four documents in the Documentation Set will be so large, or will have to be written by different providers in different locations, to necessitate placing them in separate volumes. The Documentation Standards provide a mechanism called the ROLL-OUT concept for recording sections of a document in physically separate volumes while maintaining traceability and links back to the original document. A volume can be rolled-out from a document or from another volume (i.e., volumes can be subsets of documents, or subsets of other volumes). In either case, the document or volume from which another volume is ROLLED-OUT is referred to as the PARENT (document or volume).

Within the Documentation Standards, a DATA ITEM DESCRIPTION (DID) provides a blueprint from which a writer can create one of the four documents in the Documentation Set, or any volume that is to be rolled-out from a document or other volume.

To ensure that all volumes created from the DIDs follow the same structured format making them consistent with one another, the Standards provide a tool called a TEMPLATE. The TEMPLATE provides a common place for sections of documentation that are common to all DIDs. These common sections occur at the beginning and end of the DIDs. (The content in the middle of the DID is what distinguishes one DID from another.) Examples of common sections include:

- Introduction
- Related Documentation
- Abbreviations and Acronyms
- Glossary
- Notes
- Appendices

A tailoring mechanism, which can be used for small projects, is called the IN-LINE concept. (It is the opposite of the Roll-Out concept.) Using the In-Line concept, a writer can incorporate the contents of several lower level DIDs into a single document.

STRUCTURE OF THE STANDARDS

The Standards themselves have been written using the in-line and the roll-out concepts. The parent document titled Information System Life-Cycle and Documentation Standards contains the Life-Cycle Standard in-line, while the four Documentation Standards have been rolled-out into separate volumes. For convenience of use, the Standards have been packaged as five separate small books.

Each of the four Documentation Standards contains a set of DIDs and the necessary structural guidance on how to use them. Additionally, each Documentation Standard contains sample outlines as appendices which should further help introduce the user to the structure of documents written to the Standards.

APPLICATION

This section will describe how to use the Information System Life-Cycle and Documentation Standards to create documents for a NASA project.

Using the Life-Cycle Standard

For information specifically concerning the life-cycle, the parent document, the Life-Cycle Standard should be used. It contains a separate section for each of the life-cycles: information system, hardware, software, and operational procedures. It also contains direction for when in the life-cycle the sections of documents in the Documentation Set should be written, updated, or reviewed. It also defines the activities of management, engineering, and assurance organizations during each phase of the life-cycle. Finally, it contains information on how to tailor the standard life-cycle to the nuances of a particular project. For example, guidance is given for employing evolutionary acquisition, phased delivery, incremental development, and prototyping techniques.

Using the Documentation Standards - A Sample

The Documentation Standards are tailorable to the project application size, from a single physical volume to a multi-volume set for each document type. For information concerning the preparation of a specific document in the Documentation Set, one would refer to the volume of the Standards for the type of document being created. (In the following discussion, assume that the Product Specification Documentation Standard and DIDs volume will be used to create a Software Product Specification.)

After reading the concept, rule, and guideline information in Sections 1 through 6 of the volume, one would begin by turning to the Software Product Specification Sample Outline, (provided in the Standard as Appendix C), for an overview of the document. Next, one would go to the highest level DID available--in this case, the Software Product Specification DID. Lower level DIDs will be used when more detail and substructure is required.

DID Substructure - The DID Numbering System

To help the user implement DID substructure, the Standards provide an alphanumeric numbering system. Within this system, the first letter is always either M, P, A, or R (M for management plan, P for product specification, A for assurance specification, and R for report). The next three numeric digits indicate substructure. A zero in all three digits represents the highest level document possible (i.e. one of the four documents in the Documentation Set: Management Plan, Product Specification, Assurance Specification, or Management Control and Status Reports). Subsequent levels of substructure are shown by first the hundreds digit, next the tens digit, and finally the units digit.

A two-letter suffix is added when the DID applies only to an information system or component. A DID that applies solely to an information system is identified by "-SY". A DID that applies equally well to hardware, software, or operational procedures components is identified by "-CO". Where the DID applies to only one type of component, "-HW" is used for hardware, "-SW" for software, and "-OP" for operational procedures. The absence of a suffix indicates that the DID can apply "across the board" to information systems, hardware, software, or operational procedures.

Table 1 shows how the DIDs are applied in a structured manner to a Software Product Specification. P000-SW is the highest level DID. P100, P200-SW, P400, P500, and P600-SW are subsets of P000-SW. Note that the system is not completely sequential. On Table 1, there is no P300 because P300-SY is solely for an information system and Table 1 applies to a software product specification. P310-SW and P320-SW perform the equivalent function of P300-SY for the software component. Continuing, P210 is a subset of P200-SW, and P311-SW is a subset of P310-SW. Finally, P321-SW and P322-SW are subsets of P320-SW. To summarize the logic behind the DID numbering system, hundreds digits are subordinate to triple zeroes, tens digits are subordinate to hundreds digits, and units digits are subordinate to tens digits. For further clarity, indentation is used to show levels of substructure.

When DID substructure is desired, the Standards use the roll-out and in-line concepts, along with the template as tailoring devices. Figure 1 shows a graphic example of how DIDs and the template are applied when a document is rolled-out into lower level volumes. Figure 2 shows a contrasting example of how a document is created using the substructure from the more detailed DIDs in-line, incorporating them into a higher level document.

Structure Decisions

It is the responsibility of the manager to define the life-cycle adaptation for the information system or component based on the SMAP Life-Cycle Standard. The manager must decide whether the information system is complex enough to require system decomposition into subsystems and/or components. If so, it must be determined whether each element of decomposition should have its own life-cycle and associated Documentation Set.

The manager also defines the type and amount of information to be placed in each of the four documents of the Documentation Set. For each section of each document, the manager decides whether: (1) whether information is placed in-line in that section (2) whether the section is marked "not applicable" (3) whether the information for that section is rolled-out into a physically separate volume. As each document may consist of one or more volumes, the manager defines the number of volumes to be produced for each document by section roll-out decisions.

TABLE 1

COMPLETE DID SET FOR A SOFTWARE PRODUCT SPECIFICATION

SMAP-DID-P000-SW	Software Product Specification DID
SMAP-DID-P100	Concept DID
SMAP-DID-P200-SW	Software Requirements DID
SMAP-DID-P210	External Interface Requirements DID
SMAP-DID-P310-SW	Software Architectural Design DID
SMAP-DID-P311-SW	Software External Interface Design DID
SMAP-DID-P320-SW	Software Detailed Design DID
SMAP-DID-P321-SW	Software External Interface Detailed Design DID
SMAP-DID-P322-SW	(Software) Firmware Support Manual DID
SMAP-DID-P400	Version Description DID
SMAP-DID-P500	User's Guide DID
SMAP-DID-P600-SW	Software Maintenance Manual DID

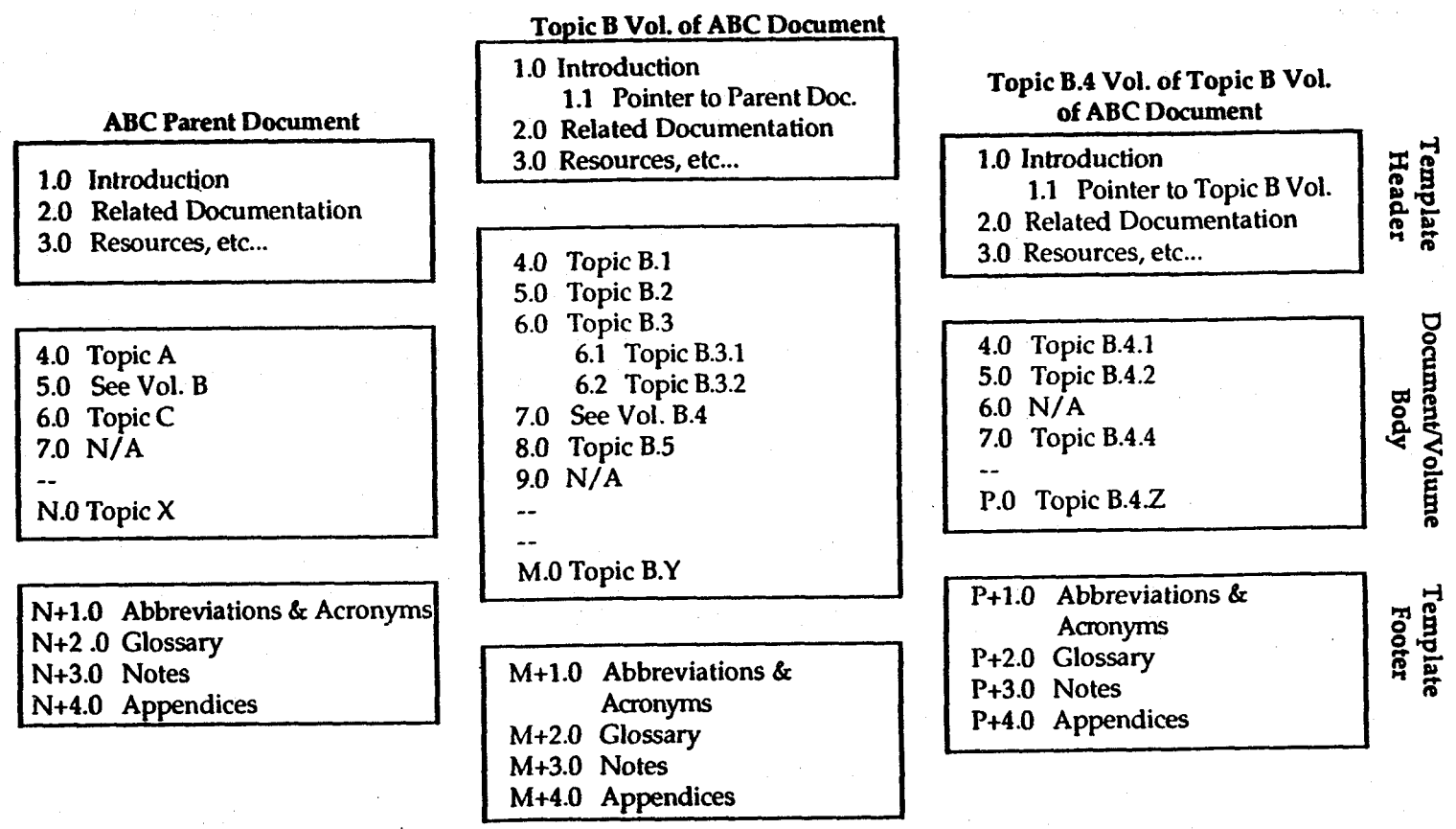


FIGURE 1 Documentation Tailoring - Roll-Out Example.

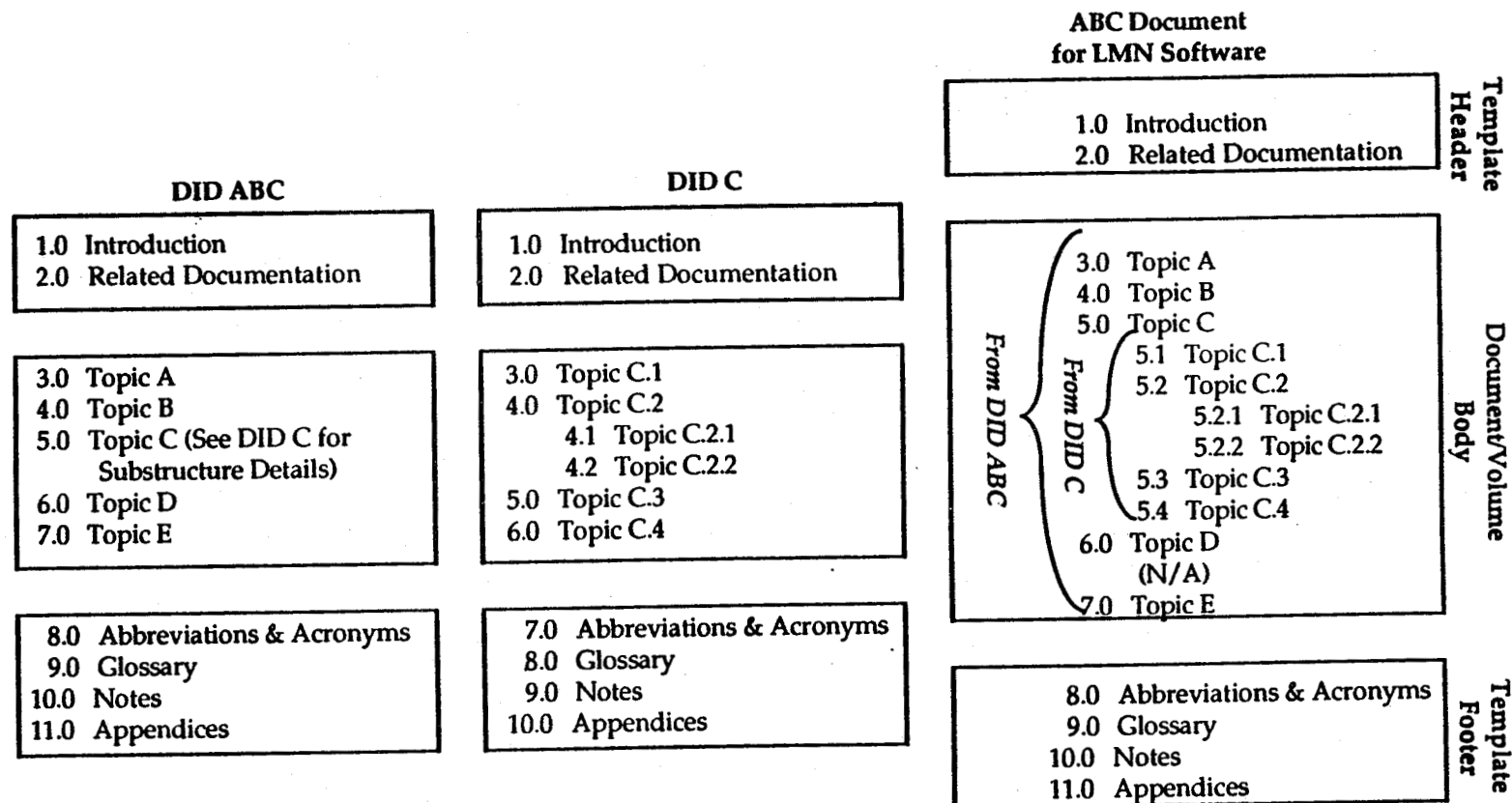


FIGURE 2

Incorporating DID Substructure In-Line.

NASA SOFTWARE ACQUISITION LIFE CYCLE

